

# PCB 생산의 자동화를 위한 데이터 변환 인터페이스 설계

## Data Transformation Interface Design for Automation of PCB Product

이승혁, 한정수  
천안대학교 정보통신학부

Lee Seung-Hyuk, Han Jung-Soo  
Division of Information & Communication,  
Cheonan Univ.

### 요약

본 연구는 PCB 생산 자동화를 위한 데이터 변환 인터페이스를 설계하였다. CAD로 설계한 PCB 부품 정보를 분석하여 Human error 검출 알고리즘을 개발하였고, IC 부품들에 관한 정보를 DB로 구축하였으며 데이터호환을 위한 알고리즘과 인터페이스를 설계하였다. 기존의 수작업 공정을 자동화함으로써 1-2일 정도의 시간을 수분 이내로 단축시킬 수 있도록 하였다.

### Abstract

In this paper, we designed data transformation interface for automation of PCB product. For this, information of PCB components designed to CAD is analyzed and human error detect algorithm is developed. Also we constructed information of IC components in database and designed the interface and algorithm for data transformation. Therefore did so that can shorten time of degree to be 1-2 days within several minutes by automating existent manual processing

## I. 서론

현재 PCB 생산 라인은 하청업체로부터 받은 CAD로 설계한 PCB 조립 데이터는 PCB 생산을 위해 Loader, Surface Mount Clean Machine, Screen Printer, Solder Print Tester, Chip Moulder, Tray Feeder 등의 설비 장비를 통하여 생산된다. 그러나 CAD로 설계한 데이터는 이들 설비 장비와 호환성이 이루어지지 않아 각 PCB에 대한 데이터를 수작업을 통하여 작업하고 있는 실정이다. 이로 인하여 수작업 후 그 정확도가 떨어져 불량 제품이 생산되는 경우가 빈번하여 많은 비용이 들고, 신뢰도가 떨어지는 경향이 있어 많은 어려움을 겪고 있다. 수작업 과정은 하나의 PCB 생산을 위한 데이터 변환만 약 8시간이 걸

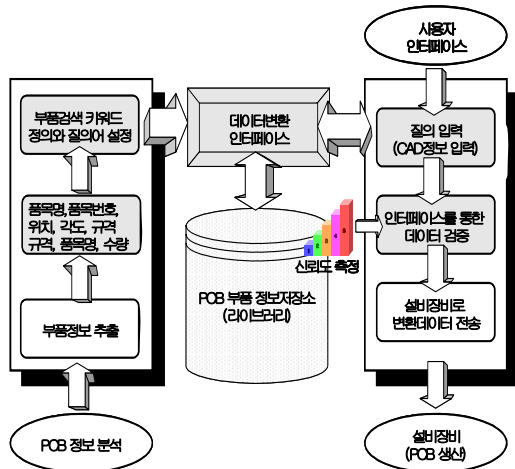
리고, 테스트 공정까지 포함하면 1-2일 정도가 걸린다. PCB 종류가 많을수록 그 시간과 비용은 기하급수적으로 증가하고, 현재 거의 대다수 전자 제품에 들어가는 PCB를 다량 생산할 수 있는 기능이 부족한 실정이다.

따라서 본 연구에서는 PCB 조립 공정에서 CAD 부품별 데이터를 분석을 통한 설비와의 데이터 호환을 위한 인터페이스를 설계, 구축하여 NC Programing 자동화로 작업능률을 극대화하기 위한 것이 그 목적이다. 이 인터페이스 구축 결과로부터 데이터 호환의 자동화가 이루어지면 서로 다른 Moulder(or Line 구성)간의 NC Program Conversion(호환)이 이루어지고, BOM ↔ NC, PGM ↔ 작업지시서 비교분석으로 Human Error들

을 사전에 제거할 수 있는 장점이 있을 뿐 아니라, Mounter Data & Line Balance 최적화로 생산성이 극대화될 수 있다는 결론을 얻을 수 있다. 또한 장착 시뮬레이션, Off-Line Teaching 으로 장착 Error의 예방이 가능하며 이로 인한 PCB의 다량 생산이 가능하고, 보다 정확한 PCB 생산과 높은 신뢰도를 얻을 수 있다.

## II. 데이터변환 인터페이스 설계

### 1. 데이터변환 시스템 구조



▶▶ 그림 1. 데이터변환 시스템 구조

CAD로 설계한 데이터들은 각각의 부품들이 보드 안에서 어느 부품이 어느 위치에 어느 각도로 접속되어야 하는지의 정보를 갖고 있다. 이때 PCB 제작 기계에는 이 정보가 호환성을 갖고 바로 연결이 되어있지 않기 때문에 PCB 제조업체는 이러한 정보를 다시 수작업으로 하나하나 모두 작성해야하는 부담을 안고 있다. 따라서 그림1에서처럼 먼저 CAD 정보 분석을 통해 부품 정보를 추출하고 PCB 생산 라인의 전 과정을 분석하여 필요한 PCB 부품에 대한 정보를 획득하며 부품별로 데이터를 분류한다. 분류된 각 PCB 부품들이 CAD로 설계한 정보와의 관계를 명확히 계

산하여 얻은 정보들을 이용하여 PCB 정보저장소를 구축한다. 저장소에 구축될 정보 안에는 품목명, 품목번호, 위치, 각도, 규격, 수량 등의 데이터들이 포함된다. 저장소는 인터페이스를 통한 부품 검색을 위해 키워드를 정의하고 데이터 변환 프로그램을 위한 준비를 완료한다. 한편, 구축된 저장소의 정보는 Human Error에 의해 잘못 계산된 값이 있을 가능성이 있다. 왜냐하면 CAD로 설계한 데이터들이 CAD 설계자의 잘못에 의해 잘못된 데이터를 가질 수 있기 때문이다. 이를 해결하기 위해 저장소에 저장된 데이터에 대해 Human Error 검증 알고리즘을 이용하여 저장 데이터를 검증 받는다. 데이터 변환 인터페이스에는 데이터 변환 알고리즘을 이용하여 CAD의 데이터를 PCB 제작 기계와 호환성을 갖는 데이터로 변환하여 사용자 인터페이스를 통하여 저장소에 저장된다. 이외에도 검증 알고리즘을 이용하여 PCB 데이터로 변환이 안정적인지 검증받게 된다. 변환 알고리즘과 테스트 알고리즘에 의해 변환된 데이터들은 PCB 생산 라인에 적용한다.

[표 1] PCB 부품 저장소의 데이터 구조

Table	Field name	Data Type	설명
1	Specification	String	저항
	Q'ty	long	수량
	Location No	String	위치
	Approved Vendor 1 st	String	
	Approved VendorParts No 01	String	
	Approved Vendor2 nd	String	
	Approved VendorParts No 02	String	
2	PartType	String	부품
	RefDes	String	위치
	PartDecal	String	부품
	Pins	long	핀수
	Layer	bool	부품
	Orient	long	각도
	X	long	좌표
	Y	long	좌표
	SMD	bool	부품
Glued	bool	부품	

## 2. PCB 부품저장소

PCB 부품저장소에는 부품 위치, 각도(X,Y 값), 규격, 품목 명, 품목 번호, 구성 수량 등의 정보로 구성된다. 이들 정보를 바탕으로 PCB 설비 장비가 인식할 수 있는 데이터들로 변환하게 되는 데이터의 구조는 표1과 같다. 또한 인터페이스를 통하여 변환된 데이터들이 시뮬레이션을 통하여 원래의 목적에 맞도록 잘 되었는지를 테스트할 수 있는 검사 프로그램을 삽입하여 설비 장비에 입력되기 전에 테스트한다. 이 결과 장비에 맞는 데이터로의 변환이 완성되면 PCB 생산 과정은 수분 이내에 자동화할 수 있다.

## 3. 데이터 정렬과 오류 검출

CAD 데이터를 추출하여 PCB 데이터로의 변환을 위해서는 CAD 데이터의 오류 검출과 해결이 반드시 필요하다. CAD 데이터는 CAD 설계자의 의해 만들어지고 설계자는 Human Error를 발생할 가능성이 매우 높기 때문이다. 이를 위하여 몇 가지 검증 알고리즘을 사용하는데 이 알고리즘은 CAD 데이터에서 품목명, 품목번호, 위치, 각도, 규격, 수량 등이 PCB 생산 설비 장비에 이미 저장되어 있는 품목명, 품목번호, 각도, 규격 등의 정보와 일치 유무를 파악하여 이것이 일치하지 않을 때는 Human Error로 간주하여 사용자에게 알려 수정을 위한 정보를 입력 받는다. 또한 CAD 데이터 내에 품목명과 품목번호가 일치하지 않거나, 수량과 PCB 보드가 수용할 수 있는 수량이 초과할 때 등의 여러 가지 잘못된 데이터들을 검출하여 새로운 값들을 입력 받는다. 이 입력된 값들 또한 잘못된 값일 가능성이 있기 때문에 다시 데이터 오류 검출 알고리즘을 통하여 정확한 데이터인지 검증받아 이 데이터들이 오류를 포함하지 않으면 다음 단계인 변환 알고리즘을 사용하여 데이터를 변환하게 된다. 그림2는 데이터 오류 알고리즘을 보여준다.

먼저 DB에 데이터들을 저장한 다음 각 item별로 분류한다. Location\_Number(위치 수)의 경우 한 Database의 field안에 여러 개의 위치 정보를 포함하

고 있다. 따라서 이를 각각의 위치 정보를 분류하여 새로운 field안에 저장하여 Q'ty(부품 수)와 Location\_Number(위치 수)가 일치하는지 확인한다. 만약 일치 하지 않다면 새로운 Q'ty(부품 수)와 Location\_Number(위치 수)를 입력받는다. 그림3은 Database에 저장된 Location\_Number 데이터들을 정렬하여 새로운 Database로 변환된 모습을 보여주고 있다. 마지막으로 item의 개수와 총 부품의 개수를 비교한다.

```

Input CAD data in Database
Database field are item_name, item_number,
                location_NO, coordinates, amount
save each filed of location_NO data
For( All Q'ty )
    if( # of Q'ty same Location_Number)
        make records for # of Q'ty in database
        separate Location_NO in each number
Input standard radian
FOR(all items)
    IF(item_number != amount)
        output ("Human Error")
        input new item_number
    ENDFIF
    
```

▶▶ 그림 2. 데이터 오류 검출 알고리즘



▶▶ 그림 3. 정렬되어 Database에 저장된 데이터

#### 4. 데이터 변환

CAD 데이터는 이미 PCB 생산 라인의 전 과정을 분석한 PCB 부품에 대한 정보와 부품별 데이터를 통해 변환 알고리즘을 이용하여 CAD 데이터를 변환한다. 그 변환 알고리즘으로 우선 Database의 두개의 Table안에 저장되어 있는 Location No와 RefDes, X, Y, Orient를 JOIN 연산에 의해 하나의 Table로 만들어야 한다. 이는 최종 PCB 생산라인에 저장해야 할 데이터가 하나의 파일만을 인식하기 때문이다. 따라서 이를 해결하기 위해 Table1의 Location No와 Table2의 RefDes가 같은 위치 정보를 가지고 있는지 확인하고 같은 위치 정보일 경우 JOIN 연산에 의해 하나의 Table로 만드는 것이다.

이렇게 하나의 Table로 저장된 데이터에서 PCB 부품별 데이터는 CAD와 같은 데이터들을 포함하고 있지만 CAD 데이터의 좌표값(x,y)과 PCB 데이터의 좌표값(x,y)이 서로 다른 체계로 되어있기 때문에 각각의 좌표값(x,y)이 어떤 위치를 중심으로 만들어졌는지 확인하고 CAD 데이터의 좌표값(x,y)을 변환하여 PCB 데이터와 호환성을 갖도록 하여야 한다. 이를 위하여 인터페이스 사용자는 각각의 좌표값(x,y)에 대한 오차를 인터페이스를 통해 입력하게 되고 이 입력된 값은 변환 알고리즘을 통해 계산되어 데이터베이스에 저장하게 된다.

또한 CAD 데이터의 부품 각도(CAD\_data Radian)는 PCB 생산 라인에서 사용하는 부품 각도(Real Radian)와 다른 경우가 발생한다. 이는 CAD 작성자가 PCB 생산 라인의 부품 각도를 분석하지 않고 단순히 부품 표준 각도에 의해 CAD를 작성하기 때문에 발생한다. 따라서 CAD 데이터의 부품 각도와 PCB 생산라인의 부품 각도를 비교하여 이를 수정하여 변환하게 된다. 그림4는 변환과정의 알고리즘이다.

```

Location No save in Database Table1
RefDes, X, Y, Orient save in Database Table2
FOR(Last Recode)
IF (Location No of Table 1 =RefDes of Table 2)
Table 1 JOIN Table2
END IF
Input standard (X,Y)
conversion (X,Y)
IF (CAD_data(radian) != Real Radian)
input new standard Radian
Do radian transformation
ENDIF

```

▶▶ 그림 4. 데이터 변환 알고리즘

#### 5. 인터페이스

데이터베이스에 저장된 데이터들이 사용자에게 의해 검증 받기 위하여 데이터베이스에 접근하여 확인받을 수 있는 인터페이스의 설계는 필수적이다. 이는 CAD 설계자의 잘못된 설계에 의한 Human Error를 미연에 방지하여 PCB 데이터로의 변환에 있어 잘못된 정보를 막을 수 있을 뿐만 아니라, 이 잘못된 정보로 인해 이후 PCB 데이터로 이용된다면 되돌릴 수 없는 막대한 손실을 얻을 수 있기 때문에 데이터베이스에 접근할 수 있는 인터페이스가 필요하다.

따라서 본 연구에서는 네트워크나 인터넷이 특별히 필요 없는 관계로 PCB 제작자의 편의성을 고려하여 운영체제는 윈도우를 이용하였으며, 데이터베이스에 접근할 수 있는 인터페이스의 개발을 위해 데이터베이스는 Access를 이용하여 OLE-DB로 데이터베이스 응용 프로그램을 만들어 접근하였다. OLE-DB는 COM 서버의 기능을 이용하여 COM 클라이언트 프로그램을 만들 것이므로 MFC보다는 ATL을 이용하는 것이 효율적이지만 ATL이 사용자 인터페이스에 관련된 기능을 거의 지원하지 않는 관계로 MFC와 ATL(Active Template Library)을 모두 이용하였다. 그림5는 인터페이스의 초기화면을 보여주고 있다.

Table 1 of Database

Specificat...	O...	Loc...	List	Pa...	2...	Par...
1608.J.1/16...	5	R15...	S...	R...	P...	2/3...
2012.J.1/40...	2	R22...	S...	R...	P...	2/3...
1608.J.1/16...	5	R22...	S...	R...	P...	2/3...
1608.F.1/1...	1	R136...	S...	R...	P...	2/3...
1608.J.1/16...	2	R10...	S...	R...	P...	2/3...

Table 2 of Database

Part...	Ref...	Par...	P	L...	O...	X	Y	S...	G...
1608...	R30	160...	S	T...	2...	-1	-1	-1	0
1608...	R30	160...	T...	2...	-1	-1	-1	0	0
1608...	R30	160...	T...	2...	-1	-1	-1	0	0
1608...	R30	160...	T...	2...	-1	-1	-1	0	0
1608...	R30	160...	T...	2...	-1	-1	-1	0	0

▶▶ 그림 5. 인터페이스 초기화면

### 참고문헌

- [1] C.Batini, s. Ceri, and S.B. Navathe, "Conceptual Database Design : An Entity-Relationship Approach", Benjamin Cummings, 1992.
- [2] K.El Emam and N.H. Madhavji "A Field Study of Requirements Engineering Practices in Information Systems Development", York, England, 1995.
- [3] 송영재, "객체지향모델링과 CBD중심 소프트웨어공학", 이한출판사, 2004.

### III. 결 론

본 연구의 목적은 그동안 수작업으로 1-2일의 시간이 소비되는 과정을 수분 이내로 단축시킬 수 있고, 직접 눈으로 테스트하는 과정에 많은 시간과 불량 생산을 막을 수 있는 정확하고 효율적인 PCB 생산을 위한 이기종간의 데이터 호환을 통한 공장 자동화가 그 목적이다. 이를 해결하기 위해 PCB 생산 전 과정을 분석하는 한편 CAD 데이터내의 Human Error를 검출하고 PCB 생산 전 과정에서 분석한 데이터를 비교하여 최소한의 오류까지 제거하였다. 이렇게 오류를 제거한 데이터들을 변환 알고리즘을 통해 CAD 데이터와 PCB 데이터간의 호환성을 갖도록 하였고 이를 사용자 인터페이스를 통해 사용자가 쉽게 알 수 있도록 설계하였다.

산업 전반에 걸쳐 있는 수많은 기계 설비들은 그 제조회사에 따라 각각의 데이터 정보들을 사용하고 있으며 이러한 정보들은 다른 설비들과의 호환성을 갖고 있지 않기 때문에 이번 연구 과제에서처럼 데이터 변환 알고리즘을 사용하여 호환성을 갖게 한다면 현재의 비효율적이고 비합리적인 설비들과의 관계에 대해 새로운 산업 발전을 가져올 수 있을 것이다.